

# 1 論文メモ書き

## 1.1 Momentum SGD

非同期な SGD の実装は”A lock-free approach to parallelizing stochastic gradient descent”などで検討されていて、実装が簡単.

$\theta$  を全てのスレッド間で共有するパラメータベクトル,  $\Delta\theta_i$  を  $i$  番目のスレッドによって計算された  $\theta$  の勾配とする.

各スレッド  $i$  はモメンタム項の更新  $m_i = \alpha m_i + (1 - \alpha)\Delta\theta_i$  とパラメータの更新

$$\theta \leftarrow \theta - \eta m_i$$

をロックなしで独立して行う.

ここで、各スレッドは独自の勾配とモメンタムベクトルを保持している.

## 1.2 RMSProp

非同期な RMSProp に関する研究はあまりされていない.

標準的な RMSProp の更新式は、以下で与えられる.

$$g = \alpha g + (1 - \alpha)\Delta\theta^2$$

$$\theta \leftarrow \theta - \eta \frac{\Delta\theta}{\sqrt{g + \epsilon}}$$

非同期で RMSProp を適用するには、要素ごとの  $g$  の移動平均をスレッドごとに共有するかどうかを決定する必要がある.

ここで、2つのパターンで実験する. 1つは、各スレッド毎に  $g$  を保持する. もう1つは Shared RMSProp と呼ばれ、ベクトル  $g$  はスレッド間で共有され、非同期に更新される. スレッド間で共有することで、メモリを削減できる.

## 1.3 実験設定

- スレッド数 : 16
- 5 回行動するごとにパラメータを更新
- shared target network は 40,000 フレームごとに更新
- Atari の実験では、”Human-level control through deep reinforcement learning”と同じ入力の前処理を用いる
- 行動繰り返し数 : 4

- ネットワークの構造は”Playing atari with deep reinforcement learning”と同じものを用いる（ストライド4の $8 \times 8$ のフィルターを16個有する畳み込み層，ストライド2の $4 \times 4$ のフィルターを32個有する畳み込み層，256個のユニットを持つ全結合層から成る．各層の活性化関数は ReLU.）
- 価値関数法では，状態-行動の価値が出力となる．
- Actor-Critic では，各行動を選択する確率（softmax）と，状態の価値の2つの出力を持つ．
- 割引率  $\gamma$  : 0.99
- 減衰係数  $\alpha$  : 0.99
- 価値関数法では， $\epsilon$  を3つの値  $\epsilon_1, \epsilon_2, \epsilon_3$  からそれぞれ確率 0.4, 0.3, 0.3 で確率的に決定
- $\epsilon_1, \epsilon_2, \epsilon_3$  は，はじめの400万フレームに渡って1.0からそれぞれ0.1, 0.01, 0.5に線形減少させる．
- A3C は， $\beta = 0.01$  のエントロピー正則化を用いた
- 初期学習率は  $\text{LogUniform}(10^{-4}, 10^{-2})$  からサンプルし，学習中に0.0に線形減少させる．